Tracking logs

- Tracking logs are made available as separate tar files on S3 in the course-data bucket.
- They are represented as JSON files that catalog all user interactions with the site.
- To avoid filename collisions the tracking logs are organized by server name, where each directory corresponds to a server where they were stored.

Common Fields

field	details	type	values/format
`username`	username of the user who triggered the event, empty string for anonymous events (not logged in)	string	
~~host~~	~~server name the user was visiting~~		
`session`	key identifying the user's session, may be undefined	string	32 digits key
`time`	GMT time the event was triggered	string	`YYYY-MM-DDThh:mm:ss.xxxxxx`
`ip`	user ip address	string	
`agent`	users browser agent string	string	
`page`	page the user was visiting when the event was generated	string	`$URL`
`event_source`	event source	string	`browser, server`
`event_type`	type of event triggered, values depends on event_source	string	*more details listed below*
`event`	specifics of the event (dependenty of the `event_type`)	string/json	*the event string may encode a JSON record*

Event Sources

The `event_source` field identifies whether the event originated in the browser (via javascript) or on the server (during the processing of a request).

Server Events

event_type	event fields	type	values/format	details	example
`show_answer`	`problem_id`	string		id of the problem being shown	"i4x://MITx/6.00x/problem/L15:L15_Problem_2"
`save_problem_check`	`problem_id`	string		id of the problem being shown	
	`success`	string	correct, incorrect	whether the problem was correct	
	`attempts`	integer		number of attempts	
	`correct_map`	string/json		*see details below*	

	state	string/json		current problem state	
	answers	string/json		students answers	
reset_problem	problem_id	string		id of the problem being shown	
	old_state	string/json			
	new_state	string/json			
$URL	GET	string/json		GET request parameters	
	POST	string/json		POST request parameters	

key	correct_map fields	type	values/format	null allowed?	details
$problem_id	hint	string			
	hintmode	boolean		yes	
	correctness	string	correct, incorrect		
	npoints	integer		yes	
	msg	string			
	queuestate	string/json	keys: key, time		

Browser Events

event_type	event fields	type	values/format	details	example
book	type	string	gotopage		
	old	integer	$PAGE	from page number	2
	new	integer	$PAGE	to page number	25
book	type	string	nextpage		
	new	integer	$PAGE	next page number	10
page_close	*empty*	string		*'page' field indicates which page was being closed*	
play_video	id	string		edX id of the video being watched	"i4x-HarvardX-PH207x-video-Simple_Random_Sample"
	code	string		youtube id of the video being watched	"FU3fCJNs94Y"
	currentTime	float		time the video was paused at, in seconds	1.264
	speed	string	0.75, 1.0,	video speed being played	"1.0"

			1.25, 1.50		
`pause_video`	`id`	string		edX id of the video being watched	
	`code`	string		youtube id of the video being watched	
	`currentTime`	float		time the video was paused at	
	`speed`	string	0.75, 1.0, 1.25, 1.50	video speed being played	
`problem_check`	*none*	string		event field contains the values of all input fields from the problem being checked (in the style of GET parameters (key=value&key=value))	
`problem_show`	`problem`	string		id of the problem being checked	
`seq_goto`	`id`	string		edX id of the sequence	
	`old`	integer		sequence element being jumped from	3
	`new`	integer		sequence element being jumped to	5
`seq_next`	`id`	string		edX id of the sequence	
	`old`	integer		sequence element being jumped from	4
	`new`	integer		sequence element being jumped to	6